International Journal of scientific research and management
(IJSRM) ||Volume||2||Issue||12||Pages|| 1750-1752 ||2014||
Website: www.ijsrm.in ISSN (e): 2321-3418

Hadoop an Emerging solution for big data

Arjunsingh R. Hajari[1], Balasaheb B. Hiwarde[2], Shubhashree savant[3]

[1]Student, MCA Department, MIT(E),
Dr. Babasaheb Ambedkar Marathwada University Aurangabad
arjun.hazari@gmail.com

[2]Student, MCA Department, MIT(E),
Dr. Babasaheb Ambedkar Marathwada University Aurangabad
hiwardebal@gmail.com

[3]Assistant professor, MCA Department, MIT(E),
Dr. Babasaheb Ambedkar Marathwada University Aurangabad
shubhashree.savant@mit.asia

Abstract: *In Today's fast growing world the needs and the technology is changing day by day for the capacity of data storage. If the storage increases the processing speed should also gradually increase. With the same objective the author has given the overview and subjective evaluation study of how the various technologies and frame work would help the big data storage.*

Keywords: Big-Data, Hadoop, Map-Reduce, HDFS ,Data-node.

1. Introduction

A way of storing enormous data sets across distributed clusters of servers and then running "distributed" analysis applications in each cluster. It's designed to be robust, in that your Big Data applications will continue to run even when individual servers — or clusters — fail. And it's also designed to be efficient, because it doesn't require your applications to shuttle huge volumes of data across your network.

What Apache says:-The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures [1].

Apache-Hadoop!:-A Solution for Big Data! .Hadoop is an open source software framework that supports data-intensive distributed applications. Hadoop is licensed under the Apache v2 license. It is therefore generally known as Apache Hadoop. Hadoop has been developed, based on a paper originally written by Google on MapReduce system and applies concepts of functional programming. Hadoop is written in the Java programming language and is the highest-level Apache project being constructed and used by a global community of contributors. Hadoop was developed by Doug Cutting and Michael J. Cafarella. And just don't overlook the charming yellow elephant you see, which is basically named after Doug's son's toy elephant [2] .

2. Architecture of Hadoop

Hadoop implements a master/slave architecture. A master node contains a JobTracker, TaskTracker, NameNode and DataNode service. It is possible to run these services on separate nodes. A slave or worker node contains a TaskTracker and DataNode. It is possible to have slave nodes that only run tasks or provide data support but this tends to be non-standard configuration.
Basically divided in two forms

2.1 Data Processing Framework and map-reduce

The data processing framework is the tool used to work with the data itself. By default, this is the Java-based system known as Map Reduce. You hear more about Map Reduce than the HDFS side of Hadoop for two reasons as It's the tool that actually gets data processed and It tends to drive people slightly crazy when they work with it.In a "normal" relational database, data is found and analyzed using queries, based on the industry-standard Structured Query Language (SQL). Non-relational databases use queries, too; they're just not constrained to use only SQL, but can use other query languages to pull information out of data stores. Hence, the term NoSQL. But Hadoop is not really a database: It stores data and you can pull data out of it, but there are no queries involved - SQL or otherwise. Hadoop is more of a data warehousing system - so it needs a system like Map Reduce to actually process the data. Map Reduce runs as a series of jobs, with each job essentially a separate Java application that goes out into the data and starts pulling out information as needed. Using Map Reduce instead of a query gives data seekers a lot of power and flexibility, but also adds a lot of complexity. There are tools to make this easier: Hadoop includes Hive, another Apache application that helps convert query language into Map Reduce jobs, for instance. But Map reduces complexity and its limitation to one-job-at-a-time batch processing tends to result in Hadoop getting used more often as a data warehousing than as a data analysis tool.

2.2 HDFS and its working

The HDFS has master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and

directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

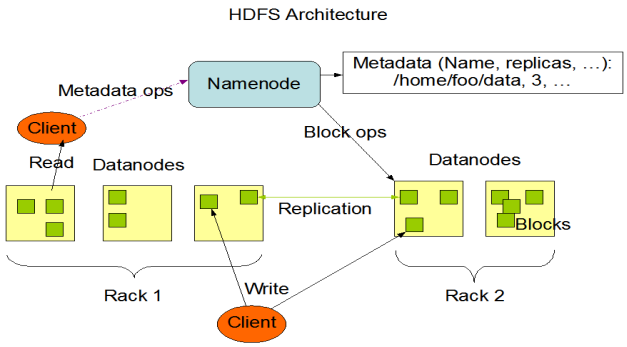

Figure 1: HDFS Architecture.

The NameNode and DataNode are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS). HDFS is built using the Java language; any machine that supports Java can run the NameNode or the DataNode software. Usage of the highly portable Java language means that HDFS can be deployed on a wide range of machines. A typical deployment has a dedicated machine that runs only the NameNode software. Each of the other machines in the cluster runs one instance of the DataNode software. The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment that is rarely the case The existence of a single NameNode in a cluster greatly simplifies the architecture of the system. The NameNode is the arbitrator and repository for all HDFS metadata. The system is designed in such a way that user data never flows through the NameNode[3].

Replication is also an core advantage in hdfs as it refers to keeping the same block of data on different nodes and Replication factor is the number of times we are going to replicate every single block of data in Hadoop replication factor is 3 by default.

And there cannot be same replicas on a single cluster as below.[4]

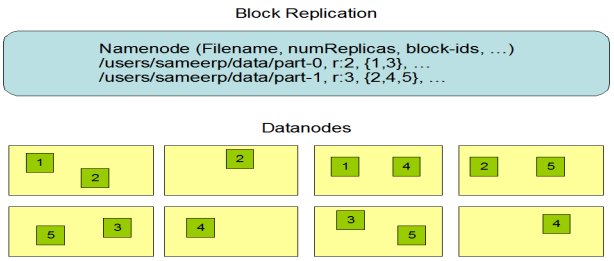

Figure 2: Data Replication.

3. What is Hadoop good for?

Searching, log processing, recommendation systems, data warehousing, video and image analysis, archiving seem to be the initial uses. One prominent space where Hadoop is playing a big role in is data-driven online websites. The four primary areas include**:-**

3.1 To aggregate "data exhaust" — messages, posts, blog entries, photos, video clips, maps, web graph.

3.2 To give data context — friends networks, social graphs, recommendations, collaborative filtering.

3.3 To keep apps running — web logs, system logs, system metrics, database query logs.

3.4 To deliver novel mashup services – mobile location data, clickstream data, SKUs, pricing[9].

4. Scenario for using Hadoop

Whenever a user a query, it isn't practical to exhaustively scan millions of items. Instead it makes sense to create an index and use it to rank items and find the best matches. Hadoop provides a distributed indexing capability.

Hadoop runs on a collection/cluster of commodity, shared-nothing x86 servers. You can add or remove servers in a Hadoop cluster (sizes from 50, 100 to even 2000+ nodes) at will; the system detects and compensates for hardware or system problems on any server. Hadoop is self-healing and fault tolerant. It can deliver data and can run large-scale, high-performance processing batch jobs in spite of system changes or failures.

Three distinct scenarios for Hadoop are:

4.1 Hadoop as an ETL and Filtering Platform – One of the biggest challenges with high volume data sources is extracting valuable signal from lot of noise. Loading large, raw data into a MapReduce platform for initial processing is a good way to go. Hadoop platforms can read in the raw data, apply appropriate filters and logic, and output a structured summary or refined data set. This output (e.g., hourly index refreshes) can be further analyzed or serve as an input to a more traditional analytic environment like SAS. Typically a small % of a raw data feed is required for any business problem. Hadoop becomes a great tool for extracting these pieces.

4.2 Hadoop as an exploration engine – Once the data is in the Map Reduce cluster, using tools to analyze data where it sits makes sense. As the refined output is in a Hadoop cluster new data can be added to the existing pile without having to re-index all over again. In other words, new data can be added to existing data summaries. Once the data is distilled, it can be loaded into corporate systems so users have wider access to it.

4.3 Hadoop as an Archive. Most of the historical data doesn't need to be accessed and kept in a SAN environment. This historical data is usually archived by tape or disk to secondary storage or sent offsite. When this data is needed for analysis, it's painful and costly to retrieve it and load it back up… so most people don't bother using historical data for their analytics. With cheap storage in a distributed cluster, lot's of data can be kept "active" for continuous analysis. Hadoop is efficient…it allows better utilization of hardware by allowing the generation of different index types in one cluster.

4.4 The entire setup is installed on top of a standalone Ubuntu machine. HADOOP requires java version 1.6 [1] or later for its proper working. HADOOP requires the use of RSA cryptography. This is accomplished with the help of ssh- keygen. The RSA key is generated for every system (i.e. both the master and the slave machines) and the key le is stored on all the other machines, in the location /home/.ssh/authorized keys. This helps, the master and slave communicate with each other without any authentication mechanism [7].

5. Companies using Hadoop an rise

Hadoop, it seems, is everywhere these days. IBM, Oracle and Yahoo are among the big guns that have been supporting Hadoop for years. Recently, Microsoft joined the club by announcing it will integrate Hadoop into its upcoming SQL Server release and Azure platforms

Microsoft's embracing of Hadoop is proof that the vendor has seen the writing on the wall about big data -- namely that it must give customers and developers the tools they need (be they proprietary or open-source) to work with all kinds of big data[6].

6. Conclusion

As the above framework will not only help the organization to expand their hands over the large data access but also reduce the cost and time complexities.
The Framework will not replace the RDBMS as it requires a lot of time and the space complexities but will definitely help to reduce the cost of the current users.
The major advantage is also the technical support which Java as it is developed in it. So Hadoop serves as a good option to invest and processed in.

7. References

[1] http://hadoop.apache.org.
[2] http://edureka.co/big-data-hadoop
[3] http://www.bigdataplanet.info
[4] htp://practicalsanalytics.wordpress.com
[5] http://www.devx.com/enterprise
[6] http://blogs.adobe.com/digitalmarketing/analytics/3-reasons-need-big-data-analytics-stratrgy
[7] N.Ramasubramanian,Shrinivas,V.V Praveen kumar Yadav "Performance evalution of stream log collection using HDFS" IJARCSSE, Volume 3,issue 6,June 2013.
[8] Jeffry Shafer,scott Rixner and Alan L.cox "The HDFS Balancing portability and performance" jeffshafer.com paper.
[9] Hadoop a definitive guide 3rd edition by Tom White.

Author Profile

Arjunsingh R. Hajari[1] has completed bachelor degree in computer application (Science) and is currently pursuing his Masters Degree (Final year) from Marathwada institute of technology (Engineering) Aurangabad. His Area of interest lies in Hadoop, DWDM and related technologies and frameworks.

Balasaheb B. Hiwarde[2] has completed bachelor degree in computer application (Mgt.) and is currently pursuing his Masters Degree (Final year) from Marathwada institute of technology (Engineering) Aurangabad. His Area of interest lies in Data warehouse and data mining and related technologies and frameworks.

Shubhashree Savant[3] is M.Sc. (Computer Science), MCA and M.Phil. (Computer Science) and is currently working in the capacity of Assistant Professor in Department of MCA at Marathwada Institute of Technology (Engineering), Dr. Babasaheb Ambedkar Marathawada University, Aurangabad, Maharashtra. As a professional member she is associated with CSI. She has total experience of 14 years in teaching. Her area of specialization includes Medical Image Processing, Database Management Systems, Network security, Cryptography and Steganography.